



UNIVERSIDAD COMPLUTENSE MADRID

Proyecto de Innovación

Convocatoria 2020/2021

Proyecto nº 18

Juez automático para el aprendizaje de bases de datos

Enrique Martín Martín

Facultad de Informática

Dpto. Sistemas Informáticos y Computación

Índice general

Índice	I
1 Objetivos propuestos en la presentación del proyecto	1
2 Objetivos alcanzados	3
3 Metodología empleada en el proyecto	5
4 Recursos humanos	6
5 Desarrollo de las actividades	7
6 Anexos	10
6.1 Listado de requisitos	11
6.2 Listado de problemas elaborados	15
6.2.1 Problemas sobre consultas SELECT	15
6.2.2 Problemas sobre modificación de bases de datos	16
6.2.3 Problemas sobre definición de funciones PL/SQL	16
6.2.4 Problemas sobre definición procedimientos PL/SQL	16
6.2.5 Problemas sobre definición de disparadores	17
6.3 Diapositivas utilizadas en el taller de difusión	18
6.4 Diapositivas utilizadas en la jornada AprendeTIC	32
6.5 Encuesta de valoración del juez por parte de los asistentes al taller	41
6.6 Resultados de la encuesta de valoración del juez	44

1. Objetivos propuestos en la presentación del proyecto

Las bases de datos son una materia básica en cualquier programa de estudios con perfil tecnológico, y como tal se enseña en varias de las asignaturas de las facultades de Informática, Ciencias Matemáticas y Estudios Estadísticos. Aunque existen varios modelos de datos para almacenar información, el más común desde los años 70 es el modelo relacional. En este modelo los datos se almacenan en tablas y se consultan usando un lenguaje muy extendido llamado SQL. Este lenguaje es aceptado por la inmensa mayoría de los Sistemas de Gestión de Bases de Datos (SGBD) que existen en la actualidad, de ahí que una de las competencias principales que se persiguen en este tipo de asignaturas sea conocer la sintaxis de SQL y ser capaz de expresar búsquedas complejas mediante consultas SQL. Aunque, estrictamente hablando, SQL es un lenguaje de consulta y manipulación de datos, tiene un gran parecido con otros lenguajes de programación y su aprendizaje es fundamentalmente práctico, así que un estudiante necesita resolver una gran cantidad de consultas antes de llegar a dominarlo. Por este motivo, el aprendizaje de SQL se podría beneficiar en gran medida del uso de sistemas automáticos de corrección de ejercicios, que permitan al alumno progresar a su ritmo en el contenido de la asignatura. Los jueces automáticos son un tipo particular de aplicaciones web que permiten a sus usuarios enviar una solución a un problema dado, realizan una serie de comprobaciones sobre dicha solución y generan un veredicto de manera automática. Su denominación proviene de su uso tradicional: servir de juez en los concursos de programación, pero se han extendido como sistemas automáticos de corrección de ejercicios.

El objetivo principal de este proyecto de innovación docente es desarrollar un juez automático que se pueda utilizar ampliamente en las asignaturas de bases de datos. Este juez automático debería tener las siguientes características:

- Cualquier persona debe poder descargarlo y ejecutarlo en sus propias máquinas.
- Cualquier persona debe poder modificarlo para adaptarlo a sus necesidades, por ejemplo para soportar nuevos SGBD.
- Debe ser posible añadir y quitar problemas en el sistema con facilidad.
- Como mínimo, el sistema debe soportar Oracle como SGBD para la corrección de ejercicios, que es uno de los principales sistemas utilizados en las asignaturas de la Facultad de Informática, aunque tendrá la posibilidad de soportar otros SGBD con facilidad.
- El sistema debe soportar problemas con consultas SQL pero también sobre creación de funciones, procedimientos y disparadores.

- La retroalimentación generada debe ser lo más informativa posible, permitiendo al estudiante entender fácilmente los errores y así poder corregirlos.

Para la consecución y planificación de este objetivo principal, se dividió en otros objetivos más pequeños:

O1) Captura de requisitos del sistema.

Los integrantes del proyecto pondrán en común ideas y propuestas sobre el juez automático, y decidirán un listado de requisitos mínimos que debe cumplir. Estos requisitos estarán divididos en dos grupos: imprescindibles y adicionales, de cara a priorizar su implementación. También se decidirá las tecnologías más adecuadas para el desarrollo del juez automático.

O2) Desarrollo del juez automático.

Se incorporarán las características detectadas en el objetivo O1 en dos fases de desarrollo y pruebas: una para las funcionalidades básicas y otra para las adicionales.

O3) Instalación y configuración de un servidor web.

Se instalará y configurará convenientemente un servidor dedicado para alojar el juez automático y permitir que cualquier usuario pueda usarlo directamente desde su navegador web.

O4) Elaboración de un conjunto de ejercicios.

Se creará un conjunto de ejercicios sobre bases de datos suficientemente amplio para utilizar en docencia y se incorporará al juez automático.

O5) Difusión de la herramienta entre profesores.

Se dará a conocer la herramienta entre los profesores interesados.

O6) Recepción de la retroalimentación del profesorado sobre la herramienta.

Se recolectarán todos los comentarios recibidos por los profesores a los que se ha presentado el juez automático.

2. Objetivos alcanzados

El objetivo principal de este proyecto ha sido alcanzado, ya que hemos desarrollado un juez automático para el aprendizaje de las bases de datos llamado «LearnSQL» con las características consideradas. Todos los sub-objetivos propuestos en la solicitud han sido alcanzados:

O1) Captura de requisitos del sistema.

Se generó un listado de requisitos que debía tener el sistema. Para la elaboración de este listado contamos con la ayuda de Tamara Huertas, Daniel Ibáñez e Iván Ruiz, estudiantes de la Facultad de Informática que contribuyeron al juez incorporando mejoras como parte de su Trabajo Fin de Grado. El listado de requisitos original que realizamos se puede encontrar en el Anexo 6.1 y su versión actualizada en <https://github.com/emartinm/lsql/blob/master/TODO.md>.

O2) Desarrollo del juez automático.

El desarrollo del juez automático se ha realizado en el repositorio público <https://github.com/emartinm/lsql/>. Todo el código fuente está disponible bajo licencia MIT (<https://opensource.org/licenses/MIT>), una de las licencias de código abierto más permisivas que existen. El juez desarrollado soporta varios tipos de problemas (consultas SQL, modificación de bases de datos, definición de funciones/procedimientos/disparadores y discriminación de consultas), contiene elementos de ludificación como un sistema de logros o podios por problema, y tiene soporte multidioma. Además de los miembros del proyecto, en el desarrollo del sistema participaron los estudiantes Tamara Huertas, Daniel Ibáñez e Iván Ruiz.

O3) Instalación y configuración de un servidor web.

El juez automático se ha desplegado en un servidor Linux dedicado exclusivamente a este fin y financiado por este proyecto. Está disponible en <https://learn.fdi.ucm.es/sql/> para profesores y estudiantes.

O4) Elaboración de un conjunto de ejercicios.

Se crearon un total de 51 ejercicios para poblar el juez, centrándose principalmente en los problemas sobre consultas SQL (39 ejercicios). En el Anexo 6.2 se puede ver un listado de los títulos de los problemas elaborados por cada tipo.

O5) Difusión de la herramienta entre profesores.

Para dar a conocer la herramienta entre los profesores más cercanos (facultades de Informática, Ciencias Matemáticas y Estudios Estadísticos) organizamos un taller online el 19 de mayo de 2021. También realizamos una presentación del juez en la jornada «Aprendizaje Eficaz con TIC

en la UCM» el 23 de junio de 2021 (AprendeTIC, <https://www.ucm.es/aprendetic/>). Las diapositivas utilizadas en el taller pueden encontrarse en el Anexo 6.3, las diapositivas utilizadas en la jornada AprendeTIC están en el Anexo 6.4 y la comunicación se podrá encontrar en las actas de la mencionada jornada.

O6) Recepción de la retroalimentación del profesorado sobre la herramienta.

Para recolectar la retroalimentación del profesorado se creó un formulario online para que los asistentes al taller pudieran hacernos llegar su opinión anónima (ver Anexo 6.5). Las respuestas obtenidas se pueden encontrar en el Anexo 6.6. Al finalizar el taller surgió un interesante intercambio de ideas, en el que aparecieron propuestas de mejoras. Dichas mejoras fueron consolidadas e integradas en el listado de requisitos de <https://github.com/emartinm/lsql/blob/master/TODO.md>.

3. Metodología empleada en el proyecto

Para la consecución del proyecto se identificaron 9 tareas; estas se realizaron sin desviaciones significativas, aunque las tareas T1–T5 se adelantaron y ampliaron para facilitar su desarrollo y ayudar a su consecución. Nos referiremos a cada miembro por sus iniciales: Iker Burgoa (IB), Pablo Cerro (PC), Jesús Correas (JC), Enrique Martín (EM), Manuel Montenegro (MM), Adrián Riesco (AR) y Rubén Rubio (RR); y el responsable de cada tarea estará resaltado en negrita. En particular tenemos las siguientes tareas:

- T1) **Captura de requisitos y especificación de tecnologías.**
Objetivos: O1. Duración: 09/2020. Miembros: IB, PC, JC, **EM**, MM, AR, RR.
- T2) **Desarrollo de un prototipo.**
Objetivos: O2. Duración: 10/2020–11/2020. Miembros: IB, PC, JC, **EM**, MM, AR, RR.
- T3) **Testing y mejora del prototipo.**
Objetivos: O2. Duración: 11/2020–05/2021. Miembros: JC, **AR**, RR.
- T4) **Desarrollo de un juez con funcionalidad completa.**
Objetivos: O2. Duración: 11/2020–05/2021. Miembros: IB, PC, **EM**, MM, RR.
- T5) **Testing y mejora del juez.**
Objetivos: O2. Duración: 11/2020–05/2021. **Miembros:** JC, AR, **RR**.
- T6) **Instalación y configuración del servidor web.**
Objetivos: O3. Duración: 01/2021. Miembros: PC, EM, **MM**.
- T7) **Elaboración de un conjunto de ejercicios.**
Objetivos: O4. Duración: 02/2021–04/2021. Miembros: **JC**, EM, MM, AR.
- T8) **Difusión de la herramienta entre profesores.**
Objetivos: O5. Duración: 05/2021. Miembros: **JC**, EM, MM, AR.
- T9) **Recepción de la retroalimentación del profesorado.**
Objetivos: O6. Duración: 06/2021. Miembros: **PC**, RR.

Coordinación

Durante la duración del proyecto se organizaron reuniones en línea para coordinar y revisar el cumplimiento de los objetivos. Además de las reuniones presenciales se utilizó un repositorio compartido y público para el código (<https://github.com/emartinm/lsql/>) y otro privado para los demás documentos (Google Drive).

4. Recursos humanos

- Enrique Martín (responsable del proyecto) ha dirigido 3 proyectos de innovación docente y ha participado en otros 5 desde 2007. En dos de esos proyectos trabajó activamente en la elaboración de un juez automático para el aprendizaje en línea (FLOP, *Free Laboratory Of Programming*) muy similar a la que se propone en este proyecto. Es profesor de la asignatura Bases de Datos desde el curso 2019-20.
- Iker Burgoa es estudiante de último curso del Grado en Ingeniería del Software, por lo que es conocedor de las dificultades de aprendizaje propias de las asignaturas sobre bases de datos. Ha colaborado en el desarrollo del juez automático como parte de su TFG.
- Pablo Cerro ha participado en 4 proyectos de innovación docente en los últimos 4 años, uno de ellos relacionado con el proyecto actual. Entre sus campos de investigación se encuentra el testeo semi-automático de aplicaciones, que se ha podido aplicar a la depuración y testeo del prototipo. Además, cuenta con experiencia en programación de aplicaciones web.
- Jesús Correas Fernández ha participado en 8 proyectos de innovación docente durante los últimos 13 años y ha dirigido uno de ellos. Ha impartido clases durante 18 años en diversas asignaturas utilizando herramientas de apoyo a la docencia, sobre las que ha dirigido varios TFG. Imparte desde hace más de un lustro las asignaturas Bases de Datos en los grados en Informática y Administración de Bases de Datos en el Máster en Ingeniería Informática. Esta experiencia ha facilitado el desarrollo del juez y de un extenso conjunto de ejercicios que cubren el temario completo de estas asignaturas.
- Manuel Montenegro ha participado en 6 proyectos de innovación docente y coordinando dos de ellos. En los últimos ocho años ha obtenido 4 evaluaciones positivas, 3 muy positivas y 2 excelentes en el programa Docencia. Ha impartido asignaturas como Ampliación de Bases de Datos y Aplicaciones Web y dirigido varios TFG con temáticas próximas a este proyecto.
- Adrián Riesco ha participado en 7 proyectos de innovación docente en los últimos 6 años y ha sido coordinador de 2 de ellos, todos relacionados con la enseñanza o la evaluación en línea. Ha dirigido recientemente múltiples TFG sobre aplicaciones web y participa regularmente en las actividades de innovación docente.
- Rubén Rubio es estudiante de doctorado en Ingeniería Informática. Como estudiante ha tenido contacto con jueces automáticos y realizó su TFG sobre un sistema semiautomático de verificación de programas. Como contratado predoctoral, ha colaborado en las clases prácticas de la asignatura Ampliación de Bases de Datos, además de en otras asignaturas y concursos de programación donde se utilizan regularmente jueces automáticos.

5. Desarrollo de las actividades

T1) **Captura de requisitos del sistema.** La captura de requisitos se realizó mediante reuniones online de los miembros del equipo, además de mediante intercambio de emails. Para esta tarea también contamos con la colaboración de Tamara Huertas, Daniel Ibáñez e Iván Ruiz, estudiantes que han colaborado con la implementación del juez automático como parte de su Trabajo Fin de Grado. Su punto de vista, junto con el de Iker Burgoa, fue muy interesante ya que propusieron ideas sobre ludificación para tratar de maximizar la participación de los estudiantes en el juez. El resultado de este intercambio de ideas fue reflejado en el listado original de requisitos que se encuentra en el Anexo 6.1, donde cada característica se etiquetó con su dificultad de integración y su importancia relativa. Una versión actualizada de este listado se encuentra en <https://github.com/emartinm/lsql/blob/master/TODO.md>.

Desde el punto de vista de tecnologías para desarrollar el juez, se decidió utilizar las siguientes:

- Python 3.9 como lenguaje de programación de todos los componentes.
- Oracle 11g como base de datos para corregir los ejercicios, pues es la que se utiliza en la asignatura «Bases de Datos» en la Facultad de Informática.
- PostgreSQL 12 como base de datos interna del juez.
- Django 3.1 o superior como *framework* para desarrollar el juez como aplicación web.
- Con respecto al resto de bibliotecas Python utilizadas, se puede encontrar un listado detallado junto con sus versiones en <https://github.com/emartinm/lsql/blob/master/requirements.txt>.
- *pylint* como analizador estático de código, para conseguir código de calidad y de aspecto uniforme.
- *GitHub* como repositorio para almacenar el código fuente.
- *Travis CI* como sistema de integración continua para realizar comprobaciones en cuanto se producían cambios en el código fuente. Entre estas comprobaciones destaca la obligatoriedad de conseguir un 100 % de cobertura de código en los tests de unidad (más detalles en <https://app.codecov.io/gh/emartinm/lsql>).

T2) **Desarrollo de un juez automático con las características básicas** y T4) **Incorporación de las funcionalidades adicionales al prototipo de juez automático.** El desarrollo del juez automático se organizó en el repositorio público <https://github.com/emartinm/lsql> con un flujo de trabajo *fork-and-branch*¹. Enrique Martín, responsable principal del código,

¹<https://blog.scottlowe.org/2015/01/27/using-fork-branch-git-workflow/>

realizaba las contribuciones directamente a la rama principal, mientras que los colaboradores externos (Iker Burgoa, Tamara Huertas, Daniel Ibáñez e Iván Ruiz) crearon un *fork* del repositorio en el que incorporaban las modificaciones en una rama (*branch*) propia y posteriormente integraban ese código en el repositorio principal mediante *pull requests*.

El juez automático con las características básicas se completó en la versión <https://github.com/emartinm/lsql/commit/4ae00f017f8374ea06e433601a94cc5ccdce93cc> el 12 de noviembre de 2020. Entre las características básicas cabe destacar:

- Soporte de problemas de consultas SQL, modificaciones DML, funciones, procedimientos y disparadores.
- Corrección de envíos con Oracle 11g.
- Generación de retroalimentación detallada.

Aunque originalmente estaba planificado para diciembre de 2020, terminar en noviembre fue muy positivo porque nos permitió probar el juez básico en un grupo de la asignatura «Bases de Datos» con resultados muy prometedores.

La incorporación de funcionalidades adicionales se adelantó a noviembre de 2020 y se simultaneó con las tareas T3, ya que consideramos que no introducía ninguna problemática adicional y sí que añadía flexibilidad a la hora de organizar la carga de trabajo. Adicionalmente, dicha tarea T4 se amplió hasta mayo para dar cobertura a todos los trabajos de Iker Burgoa, Tamara Huertas, Daniel Ibáñez e Iván Ruiz. Las funcionalidades adicionales se propusieron inicialmente a través de email entre los participantes, pero luego se procedió a identificar cada mejora como un *issue* en el repositorio (ver <https://github.com/emartinm/lsql/issues?q=is%3Aissue+is%3Aclosed>) para centralizar la discusión. La integración y revisión del código implementando cada mejora se realizó mediante *pull request* (ver <https://github.com/emartinm/lsql/pulls?q=is%3Apr+is%3Aclosed>). Entre las funcionalidades adicionales destacamos:

- Creación de clasificaciones por cada grupo de clase y colección de problemas.
- Incorporación de elementos de ludificación: logros y podios.
- Inclusión de un nuevo tipo de problema sobre discriminación de consultas.
- Soporte multidioma para español e inglés.
- Inclusión de pistas en los problemas.

T3) Prueba del prototipo básico para detectar bugs y corregirlos y T5) Prueba del juez automático para detectar bugs y corregirlos. Las tareas de prueba del sistema se ampliaron temporalmente para coincidir con todo el desarrollo del juez, en lugar de concentrarlas en un par de periodos concretos. De esta manera, la incorporación de mejoras, la detección de errores y su corrección se realizó de manera simultánea en el tiempo. Esta planificación resultó más cómoda, ya que no implicó paralizar una tarea mientras se estaba realizando la otra y además no provocó ninguna complicación adicional.

El listado de defectos detectados y su discusión se centralizó también en el repositorio GitHub mediante *issues*, tal y como se ve en <https://github.com/emartinm/lsql/issues?q=is%3Aissue+is%3Aclosed>.

- T6) **Instalación y configuración del servidor web.** La instalación y configuración del servidor web se realizó según la planificación, a finales de enero de 2021. Como el servidor Dell PowerEdge T140 adquirido con el presupuesto del proyecto no estuvo disponible hasta ese momento, se realizaron pruebas con un equipo temporal disponible en el Departamento de Sistemas Informáticos y Computación y luego se migró toda la configuración al nuevo equipo. Para la configuración del equipo se utilizó el sistema operativo *Ubuntu 20.04 LTS* junto con los servidores *NGINX* y *Gunicorn*.
- T7) **Elaboración de un conjunto de ejercicios.** Se crearon un total de 51 ejercicios para incluir en el juez, principalmente ejercicios sobre consultas SQL (39 ejercicios). En el Anexo 6.2 se puede ver un listado de los títulos de los problemas elaborados por cada tipo.
- T8) **Difusión de la herramienta entre profesores.** Para dar a conocer el juez entre los profesores más cercanos a la temática se organizó un taller online de 1:30h de duración el 19 de mayo de 2021 en la Facultad de Informática. Al finalizar el taller se produjo una discusión entre distintos participantes de la que surgieron ideas muy interesantes.
- Para difundir la herramienta también realizamos una presentación del juez en la jornada «Aprendizaje Eficaz con TIC en la UCM» el 23 de junio de 2021 (AprendeTIC, <https://www.ucm.es/aprendetic/>) y publicamos una comunicación en sus actas.
- Las diapositivas utilizadas en el taller pueden encontrarse en el Anexo 6.3, las diapositivas utilizadas en la jornada AprendeTIC en el Anexo 6.4 y la comunicación aparecerá en las actas de la mencionada jornada.
- T9) **Recepción de la retroalimentación del profesorado sobre la herramienta.** Tras la realización del taller, se realizó una encuesta entre los asistentes para conocer la opinión sobre distintos aspectos de la herramienta. Se plantearon cuestiones relativas a la usabilidad de la herramienta, la facilidad de navegación entre las distintas secciones, la funcionalidad, y la utilidad del juez en el contexto de asignaturas relacionadas con las bases de datos. El cuestionario utilizado se encuentra en el Anexo 6.5.
- Las valoraciones fueron, en general, muy positivas (ver Anexo 6.6). Entre las personas que respondieron, aquellas que impartían (o habían impartido) asignaturas relativas a las bases de datos afirmaron que utilizarían el juez en su asignatura. Entre los aspectos más positivos se destacaron la facilidad de uso, y la evaluación inmediata que guía al estudiante a la solución correcta. Por otro lado, también se aportaron algunas sugerencias de mejora. Entre ellas, mejorar la navegación entre los distintos ejercicios, incorporar más ejercicios al catálogo, e introducir casos de prueba ocultos para evitar soluciones *ad hoc* que solamente funcionen con el caso de ejemplo mostrado.

6. Anexos

6.1. Listado de requisitos

TODO:

- Hacer un robots.txt para evitar que indexen imágenes ni nada
- Indicar ilustraciones obtenidas de:
 - <https://freesvg.org/red-cross-not-ok-vector-icon> (dominio público)
 - <https://icons.getbootstrap.com/> (MIT)

Mejoras propuestas:

- (Jesús) [**Dificultad alta**] [IMPRESINDIBLE] Soporte multi-idioma para mostrar el sistema en inglés y (Daniel) otros idiomas incluidos los demás de España como el valenciano. (<https://docs.djangoproject.com/en/3.1/topics/i18n/>)
- (Jesús) [**Dificultad baja**] [IMPRESINDIBLE] Botón para descargarse el *script* de creación de las tablas y carga de datos en cada problema.
- (Jesús) [**Dificultad alta**] [INTERESANTE] Triggers: mejorar la comparación de resultados y diseño de casos de prueba.
- (Jesús y Enrique)
 - [**Dificultad media**] [IMPRESINDIBLE] Agrupar usuarios en clases
 - [**Dificultad baja**] [INTERESANTE] Almacenar puntuación para calificar automáticamente algunos ejercicios, mostrar el ranking (modo competición: ver los resultados de los demás estudiantes y sus puntos, primer estudiante en resolver cada problema) y generar un reporte (idealmente Moodle)
 - [**Dificultad alta**] [INTERESANTE] Modo examen: horario reducido, sin retroalimentación de solución o sin mostrar el resultado esperado. ¿Crear una agrupación concreta para cada examen con usuarios nuevos o reutilizar los existentes?
- (Jesús) [**Dificultad media**] Representación gráfica del diseño de BD de cada pregunta (al estilo del modelo relacional o tipo UML, representando claves externas y primarias, etc.).
- (Enrique) [**Dificultad baja**] [INTERESANTE] Mostrar al menos las tablas de manera más compacta, p.ej. mostrando solo la cabecera y que se pueda ampliar para ver el contenido pulsando en el típico “+”.
- (Jesús) [**Dificultad alta**] Mejorar retroalimentación: pistas más inteligentes, p.ej. **Te falta usar la tabla X en FROM**
- (Enrique) [**Dificultad alta**] Soporte de otros SGBD: PostgreSQL, MySQL, MariaDB, SQLite, etc.
- (Jesús) [**Dificultad media**] [INTERESANTE] Comprobar también la salida estándar en PL/SQL (PUT_LINE)

- (Jesús) [**Dificultad media**] [**INTERESANTE**] Gestión de excepciones en PL/SQL: ver que se lanzan las que deberían
- (Jesús) [**Dificultad media**] Permitir varios casos de prueba en PL/SQL con distintos valores para los parámetros (actualmente suportado en funciones pero no en procedimientos).
- (Enrique) [**Dificultad media**] Permitir varias BD iniciales como casos de prueba en los ejercicios (actualmente solo se considera una BD inicial).
- (Daniel) [**Dificultad baja**] Subir solución como fichero SQL arrastrando o con botón.
- (Jesús) [**Dificultad alta**] Diseñar los conjuntos de datos necesarios para probar una consulta que puede ser errónea. Otra enfoque (Enrique) sería dar dos consultas SQL y pedir al estudiante qué datos tendrían que tener las tablas para que los resultados fueran diferentes.
- (Daniel) [**Dificultad media**] [**INTERESANTE**] Sistema de logros para aumentar la competitividad, p-ej. **Realiza 3 ejercicios de inserción**
- (Daniel) [**Dificultad alta**] [**INTERESANTE**] Corrección diciendo si está algo ‘medio bien’, como refuerzo a las pistas.
- (Daniel) [**Dificultad baja**] [**INTERESANTE**] Añadir tanto primer alumno en resolver ejercicio, como el segundo y tercero, como si fuera un podio.
- (Daniel) [**Dificultad alta**] [**INTERESANTE**] Sistema de comunicación con el profesor para resolver dudas de ejercicios específicos.
- (Daniel) [**Dificultad media**] [**INTERESANTE**] Función para que el alumno pueda agregar algún caso de prueba específico => solo relativo a INSERT VALUE. Enrique: ojo a posibles problemas de inyección.
- (Tamara) [**Dificultad baja**] [**IMPRESINDIBLE**] En el apartado “Mis envíos” añadir un botón para descargar el código enviado de un ejercicio.
- (Tamara e Iker) [**Dificultad media**] [**INTERESANTE**]
 - Opción de pista en cada ejercicio para los alumnos que no sepan empezar un ejercicio. En el caso de tener un sistema de puntos se les restaría un porcentaje por cada ejercicio en el que han necesitado ayuda.
 - Cada alumno tiene una serie de “saldo” el cual recibe tras un aceptado, logros (subir tres seguidos bien, alguno por dificultad tener más premio), que los comentarios de ayuda aparezca un botoncito con un máximo de 3 ayudas y cada vez más caros (3 , 5 , 10 por ejemplo) y sean ayudas genéricas para cada ejercicio.
- (Jesús) [**Dificultad media**] [**INTERESANTE**] Añadir una opción que permita introducir en formato “libre” (como SQLDeveloper, o

<https://livesql.oracle.com/>) instrucciones a ejecutar en un esquema limpio de BD. Los esquemas generados de esta forma deben permitir un nivel de persistencia asociado al usuario, por ejemplo hasta las 00:00 del día siguiente. O mejor aún, que se pueda configurar para cada usuario.

6.2. Listado de problemas elaborados

6.2.1. Problemas sobre consultas SELECT

1. Ordenar resultados usando varios campos
2. Seleccionar filas
3. Filtrar columnas
4. Condiciones sobre cadenas
5. Estadísticas sobre la altura de jugadores
6. Tabla completa
7. Ordenar resultados
8. Renombrar columnas
9. Operaciones sobre columnas
10. Árbitros dirigiendo empates
11. Personas que no son jugadores
12. Ramas de patrocinadores de clubes populares
13. Clubes arbitrados por un determinado árbitro
14. Tabla completa
15. Asistentes a exactamente una victoria de equipo local
16. Partidos ganados por el club local
17. Número de jugadores por club (con nombres)
18. Estadísticas sobre financiaciones de clubes
19. Partidos dirigidos por cada árbitro, incluyendo árbitros inéditos
20. Número de clubes diferentes con jugadores patrocinados (por cada patrocinador)
21. CIF y nombre de los clubes con más de 1 jugador
22. Ordenar resultados usando funciones
23. Clubes con el máximo número de socios
24. Clubes tales que todos sus jugadores tienen la letra E
25. CIF de los clubes con más de 1 jugador
26. Seleccionar filas y columnas

27. Árbitros en cuyos partidos se meten cada vez más goles locales
28. Asistentes a partidos que son jugadores o árbitros
29. Número de jugadores por club
30. Estadísticas sobre financiaciones de clubes, incluyendo todos los clubes
31. Partidos dirigidos por cada árbitro
32. Clubes con su diferencia de socios respecto a la media
33. Personas que han asistido a victoria local
34. Clubes que han ganado todos sus partidos locales
35. Omitir duplicados
36. Jugadores que reciben patrocinio de todos los patrocinadores deportivos
37. Consulta combinada
38. Financiación total de cada club
39. Complete table

6.2.2. Problemas sobre modificación de bases de datos

1. Insertar una fila

6.2.3. Problemas sobre definición de funciones PL/SQL

1. Goles totales a partir del resultado
2. Victorias locales dirigidas por un árbitro
3. Extraer goles locales
4. Extraer partes de una cadena
5. Partidos ganados por un club

6.2.4. Problemas sobre definición procedimientos PL/SQL

1. Goles totales a partir del resultado
2. Victorias locales dirigidas por un árbitro
3. Extraer goles locales
4. Extraer partes de una cadena
5. Partidos ganados por un club

6.2.5. Problemas sobre definición de disparadores

1. Modificar las filas insertadas/modificadas

6.3. Diapositivas utilizadas en el taller de difusión



LearnSQL: un juez automático para el aprendizaje de bases de datos

Enrique Martín (emartinm@ucm.es)
Taller Fac. Informática

Contenidos

- 1 Introducción
- 2 Características del juez
- 3 Uso durante el curso 2020/21
- 4 Conclusiones

Introducción

- Los **jueces automáticos** como DomJudge se usan ampliamente en asignaturas de programación y suponen una ayuda:
 - Estudiantes: pueden practicar en cualquier momento
 - Profesores: automatizan la corrección funcional
- Sin embargo, estos jueces realizan una comparación muy estricta para decidir si un envío es correcto, además de no proporcionar apenas información de retroalimentación: *Wrong Answer* sin explicar por qué

Motivación

Nos pareció muy interesante tener un juez automático **adaptado** a la asignatura de «Bases de Datos» y que proporcionara la **máxima retroalimentación** posible

Objetivo

Características que buscamos en el juez:

- **Sencillo** de utilizar tanto para profesores como estudiantes
- **Aplicable** a la asignatura de «Bases de datos» sin esfuerzo (Oracle, tipos de problemas, etc.)
- Juez automático para el **aprendizaje**, no para la competición: dar todas las ayudas para que el estudiante entienda el error en su envío y pueda corregirlo
- **Código abierto** disponible para todos
- Abierto a la **ludificación** (logros, podios, clasificación...)
- **Transparente**: el estudiante debe poder consultar todos sus envíos

Apoyos

- Proyecto **INNOVA-Docencia** del curso 2020/21 dentro del área «ciencias e ingenierías», con un presupuesto de 1.125€ para comprar un servidor
- **Trabajo de Fin de Grado** de la Fac. de Informática en el curso 2020/21, con la colaboración de 4 estudiantes dedicados a la mejora y extensión de la funcionalidad

Integrantes del proyecto INNOVA-Docencia y del TFG:

- ① Iker Burgoa (estudiante UCM)
- ② Pablo Cerro (profesor UCM/UAM)
- ③ Jesús Correas (profesor UCM)
- ④ Tamara Huertas (estudiante UCM)
- ⑤ Daniel Ibáñez (estudiante UCM)
- ⑥ Enrique Martín (profesor UCM)
- ⑦ Manuel Montenegro (profesor UCM)
- ⑧ Rubén Rubio (profesor UCM)
- ⑨ Iván Ruiz (estudiante UCM)

Sistema

- **Código fuente** disponible bajo licencia MIT:
 - <https://github.com/emartinm/lsql>
- **Sistema desplegado** para su uso en la asignatura de «Bases de Datos»
 - <https://learn.fdi.ucm.es/sql/>

Características del juez

Todos los asistentes al taller tenéis una cuenta de estudiante en el grupo ficticio «Taller mayo 2021» con la que podréis realizar cualquier prueba:

- **URL:** <https://learn.fdi.ucm.es/sql/>
- **Usuario:** lo que haya **antes de la @** en el correo que usasteis para inscribiros (ber.toldo@ucm.es → usuario ber.toldo)
- **Contraseña:** TallerMayo2021 (podéis/debéis cambiarla una vez hayáis accedido)

Características del juez

Nos detendremos en las siguientes características del juez:

- Organización de los elementos
- Corrección de los ejercicios
- Tipos de problemas soportados
- Veredictos recibidos
- Retroalimentación detallada
- Ludificación: clasificación, podios, logros, pistas
- Soporte multidioma
- Interfaz Django para administración
- Cargar colecciones y problemas a partir de ficheros ZIP

Organización de los elementos

- Existen 2 tipos de usuarios del sistema
 - **Estudiantes:** pueden ver los problemas, realizar envíos y consultar la clasificación. Pertenecen a grupos de clase
 - **Profesores:** además de todo lo que hacen los estudiantes, pueden añadir/borrar/modificar cualquier elemento del sistema
- Organización de los problemas
 - El sistema contiene una serie de **colecciones** de problemas
 - Cada colección contiene una serie de **problemas**
 - Las colecciones y problemas son visibles y utilizables **por todos los estudiantes**, independientemente de su grupo

- Para la corrección de los envíos, LearnSQL **ejecuta el código** del estudiante en Oracle 11g usando un **esquema y usuario temporal** que es destruido una vez se ha generado el veredicto
 - Esto hace que el proceso sea algo lento (aprox. 1 segundo por comprobación) pero evita que la ejecución de un envío afecte al resto de estudiantes
- El resultado obtenido por el código del estudiante se compara con el resultado obtenido por la **solución oficial**
- Existe soporte para considerar **varias BD iniciales**, aunque actualmente está incorporado únicamente en los problemas de consultas **SELECT**
- Existen enfoques diferentes: comprobación de equivalencia semántica para Consultas Conjuntivas (SQLator, SQLify) e incluso heurísticas examinando el texto

Tipos de problemas

Para conseguir que fuera lo más útil posible en la asignatura de «Bases de datos» necesitábamos que soportase distintos tipos de problemas:

- 1 Consultas **SELECT**
- 2 Modificación de tablas con sentencias DML
- 3 Definición de funciones PL/SQL
- 4 Definición de procedimientos PL/SQL
- 5 Definición de disparadores
- 6 Encontrar los datos que hacen que una consulta devuelva valores incorrectos (problema discriminante)

Consultas SELECT

- El tipo de **problema más común**
- A partir de una BD inicial, se pide una consulta **SELECT** que devuelva los resultados esperados
- Se corrige comparando los resultados obtenidos frente a la solución oficial
- Permite comparar los resultados teniendo en cuenta el orden o no

Tabla completa

Considera una tabla que almacena algunos datos sobre clubes de fútbol definida de la siguiente manera:

```
CREATE TABLE Club(  
  CIF CHAR(9) PRIMARY KEY,  
  Nombre VARCHAR2(40) NOT NULL UNIQUE,  
  Sede VARCHAR2(30) NOT NULL,  
  Num_Socios NUMBER(10,0) NOT NULL,  
);
```

Escribe una consulta SQL que devuelve **todo el contenido** de la tabla `Club`. El esquema del resultado debe ser el siguiente:

```
(CIF, Nombre, Sede, Num_Socios)
```

Modificación de tablas con sentencias DML

- A partir de una BD inicial, se pide una serie de sentencias SQL que modifiquen tablas: insertar filas, borrar filas, actualizar filas
- Se corrige comparando que la BD resultante tras ejecutar las sentencias es la misma que se obtiene con la solución oficial

Insertar una fila

Considera una tabla que almacena algunos datos sobre clubes de fútbol definida de la siguiente manera:

```
CREATE TABLE Club(  
  CIF CHAR(9) PRIMARY KEY,  
  Nombre VARCHAR2(40) NOT NULL UNIQUE,  
  Sede VARCHAR2(30) NOT NULL,  
  Num_Socios NUMBER(10,0) NOT NULL,  
);
```

Escribe una consulta SQL que inserte un nuevo club en la tabla. La nueva fila debe contener los siguientes datos:

```
('11111114X', 'Real Betis Balompié', 'Av. de Heliópolis, s/n', 45000)
```

Enrique Martín

Taller LearnSQL

Fac. Informática - UCM

16 / 39

Definición de funciones PL/SQL

- El estudiante envía la definición de una función PL/SQL
- Se corrige comparando el resultado devuelto por la función con el valor esperado generado por la solución oficial
- Se pueden probar distintas invocaciones con parámetros diferentes

Extraer partes de una cadena

Crea una función `golesLocal` que recibe una cadena de texto `VARCHAR2` representando el resultado de un partido con el formato `golesLocal-GolesVisitante` y devuelve el número de goles del equipo local como un valor de tipo `NUMBER`. Por ejemplo, `golesLocal('3-0')` debería devolver el número `3`.

Resultado esperado

Llamada	Resultado
<code>golesLocal('0-1')</code>	0
<code>golesLocal('2-1')</code>	2
<code>golesLocal('12-13')</code>	12
<code>golesLocal('117-99')</code>	117

Enrique Martín

Taller LearnSQL

Fac. Informática - UCM

17 / 39

Definición de procedimientos y disparadores

- El estudiante envía la definición de un procedimiento o disparador
- Se realiza una invocación al procedimiento o una operación DML que lanza el disparador
- Se comprueba que los cambios producidos en la BD por la ejecución del código del estudiante son exactamente idénticos a los producidos por la solución oficial

- Tipo de problema novedoso que sirve para entrenar la comprensión profunda de las consultas SQL:
 - 1 Se proporciona una BD con datos iniciales
 - 2 Se presenta una consulta SQL que *presuntamente* resuelve una tarea concreta. **Es incorrecta**, pero para los datos iniciales devuelve el resultado esperado,
 - 3 Se pide al estudiante que inserte filas en la BD para encontrar un estado en la que la consulta SQL sí devuelve resultados erróneos
- Para corregir envíos se comprueba que con la nueva BD inicial la consulta errónea y la consulta correcta oficial sí devuelven resultados diferentes
- Puede tener en cuenta el orden en los resultados

Problema discriminante

Conteo de atletas

Queremos saber qué clubes de atletismo tienen más de **1000 atletas**, los datos están almacenados de la siguiente forma:

```
CREATE TABLE Club(
  Nombre VARCHAR2(40) PRIMARY KEY,
  Pista VARCHAR2(30) NOT NULL,
  Atletas NUMBER(10,0) NOT NULL,
  CONSTRAINT NumAtletas CHECK (Atletas >= 0)
);
```

La solución dada por el becario es la detallada en la línea de código de abajo. Creemos que esta solución no es correcta aunque el resultado sea bueno. ¿Qué dato/s deberíamos introducir para que esta sentencia no sea correcta para este caso?:

Base de datos

[Descargar script](#)

CLUB

NOMBRE	PISTA	ATLETAS
C.C. Garbí	Pista de Atletismo de Gandia	2500
CA Safor	Pista de Atletismo del Grao	2000
València Terra i Mar	Pista del Túria	5000

Consulta SQL errónea a depurar

```
SELECT * FROM Club;
```

Tipos de veredictos

El realizar un envío, el estudiante recibe un **veredicto**:

- **Aceptado**: el envío produce los resultados esperados
- **Error de validación**: el envío no se ha llegado a ejecutar porque no cumple algunos requisitos previos, p.ej., se espera 1 consulta SQL pero se envían 3 sentencias
- **Error en ejecución**: la ejecución del envío ha producido errores como tablas que no existen, errores sintácticos, claves duplicadas, etc.
- **Resultados incorrectos**: el envío se ha ejecutado sin problema pero los resultados no son los esperados
- **Tiempo limite excedido**: la ejecución tarda demasiado tiempo

Retroalimentación detallada: error en ejecución

Cuando se produce un error en ejecución se indica la **razón** tal cual la devuelve Oracle, y, si existe, la **línea** y **columna** donde se ha producido ese error:

```
1 SELECT * FROM Club WHERE Num_Socios < 5
```

Enviar solución

Retroalimentación

Posición: línea 1, columna 10
ORA-00923: FROM keyword not found where expected

Enrique Martín

Taller LearnSQL

Fac. Informática - UCM

22 / 39

Retroalimentación detallada: resultados incorrectos

Se trata de indicar exactamente qué filas son incorrectas:

```
1 SELECT * FROM Club WHERE Num_Socios < 20000
```

Enviar solución

Retroalimentación

Faltan algunas filas que deberían aparecer.

Resultado generado por tu código:

CIF	NOMBRE	SEDE	NUM_SOCIOS
11111113X	Paris Saint-Germain Football Club	Rue du Commandant Guilbaud	1000

Filas que faltan:

CIF	NOMBRE	SEDE	NUM_SOCIOS
11111111X	Real Madrid CF	Concha Espina	70000
11111112X	Futbol Club Barcelona	Aristides Maillol	80000

Enrique Martín

Taller LearnSQL

Fac. Informática - UCM

23 / 39

Clasificación

Para cada grupo de clase y colección de problemas se puede visualizar la clasificación (por número de problemas resueltos y envíos necesarios para resolver dichos problemas):

Pos.	Usuario	EJ1	EJ2	EJ3	EJ4	Puntuación	Resueltos
1	Eva 🏆x4	1/2 (2)	1/5 (5)	1/3 (3)	1/1 (1)	11	4
2	Juan 🏆x4	1/8 (8)	1/1 (1)	1/14 (14)	3/7 (3)	26	4
3	Ana 🏆x4	7/34 (17)	2/32 (15)	3/11 (8)	3/11 (8)	48	4
4	Alberto 🏆x5	1/82 (44)	1/6 (6)	1/91 (91)	1/57 (57)	198	4
5	Jacinto 🏆x1	1/3 (3)	0/0 (0)	1/2 (2)	1/1 (1)	6	3
6	Soledad 🏆x5	1/8 (8)	0/6 (6)	1/2 (2)	1/4 (4)	14	3

Enrique Martín

Taller LearnSQL

Fac. Informática - UCM

24 / 39

Podios (ludificación)

Para fomentar la participación, se muestran los 3 primeros estudiantes que resuelven cada problema (sin tener en cuenta grupos de clase)

Funciones PL/SQL

Ejercicios en los que hay que crear **funciones** PL/SQL

Listado de problemas

Nombre	Número de envíos	🏆 1º 🏆	🏆 2º 🏆	🏆 3º 🏆
✓ Extraer goles locales	0	Ana	Juan	-
✓ Goles totales a partir del resultado	2	Ana	Eva	Juan
✓ Partidos ganados por un club	2	Juan	Luis	John
✓ Victorias locales dirigidas por un árbitro	4	Eva	Pepe	Julian

⏪ ⏩ ⏴ ⏵ ⏶ ⏷ ⏸ ⏹ ⏺ ⏻ ⏼ ⏽ ⏾ ⏿ 🔍 ↺ ↻

Enrique Martín

Taller LearnSQL

Fac. Informática - UCM

25 / 39

Logros (ludificación)

- Para fomentar la participación, se pueden definir logros que los estudiantes consiguen al resolver problemas. Ej:
 - Resolver 20 problemas
 - Resolver 5 problemas de la colección «Consultas SQL»
 - Entrar en el podio de 5 problemas
- El estudiante puede consultar la página «Mis logros» para ver sus logros pendientes y los que ya ha conseguido

Página de logros de Eva

Logros conseguidos

Nombre	Descripción	Fecha
Principiante	Resuelve 5 problemas de cualquier tipo	18 de Noviembre de 2020 a las 21:29
Amigo del podio	Entra en el podio en al menos 3 problemas	18 de Noviembre de 2020 a las 21:29

Logros pendientes

Nombre	Descripción
Maestro	Resuelve 25 problemas de cualquier tipo

- Para favorecer su visibilidad, el número de logros de cada estudiante se muestra en la clasificación

⏪ ⏩ ⏴ ⏵ ⏶ ⏷ ⏸ ⏹ ⏺ ⏻ ⏼ ⏽ ⏾ ⏿ 🔍 ↺ ↻

Enrique Martín

Taller LearnSQL

Fac. Informática - UCM

26 / 39

Soporte para pistas

- Todavía en desarrollo (*pero muy avanzado*)
- Permitirá asociar a cada problema una serie de pistas
- Cada pista se podrá desbloquear al realizar un cierto número de envíos
- *Posiblemente* la visualización de pistas afectará a la clasificación

⏪ ⏩ ⏴ ⏵ ⏶ ⏷ ⏸ ⏹ ⏺ ⏻ ⏼ ⏽ ⏾ ⏿ 🔍 ↺ ↻

Enrique Martín



Taller LearnSQL

Fac. Informática - UCM

27 / 39

- La interfaz está disponible en español e inglés
- Los problemas tienen asignado un idioma (español por defecto)

The screenshot shows the 'Learn SQL' website interface. At the top, there's a navigation bar with 'Exercises', 'Ranking', and 'Help' links, along with a user profile 'usuario@ucm.es' and a language selector set to 'English'. Below the navigation bar, a dropdown menu is open, showing options: 'My submissions', 'My achievements', 'Change password', 'Judge statistics', and 'Logout'. The main content area is titled 'Problem collections' and displays a table with columns 'Name', 'Solved', and 'Total problems'. The table lists several collections, some with checkmarks indicating completion status.

Name	Solved	Total problems
Consultas SELECT básicas	3	12
Reuniones de tablas y operaciones de conjuntos	0	8
Agregaciones y agrupaciones	0	12
Consultas anidadas	1	5
✓ Funciones PL/SQL	4	4
✓ Procedimientos PL/SQL	2	2
Problemas mezclados / Mixed problems  	3	7

Interfaz de administrador Django

Un profesor puede acceder y modificar cualquier elemento del sistema desde la interfaz de administrador Django

The screenshot shows the Django Admin interface. On the left, there's a sidebar with navigation links: 'Inicio', 'Judge', and 'Select problems'. Under 'JUDGE', there are links for 'Achievement definitions', 'Collections', 'Discriminant problems', 'Dml problems', 'Function problems', 'Num solved achievement definitions', 'Num solved collection achievement definitions', and 'Num solved type achievement definitions'. The main content area is titled 'Seleccione select problem a modificar' and shows a list of problems with columns 'Acción', 'TITLE MD', 'CREATION DATE', and 'COLLECTION'. A filter sidebar on the right allows filtering by 'Por creation date' with options like 'Cualquier fecha', 'Hoy', 'Últimos 7 días', 'Este mes', and 'Este año'.

Inserción sencilla de colecciones y problemas

Los problemas y colecciones se pueden añadir fácilmente desde **ficheros ZIP**:

The screenshot shows a file explorer window titled '01_select_all_EN.zip'. It displays a list of files with columns 'Nombre', 'Tamaño', 'Tipo', and 'Modificado'. The files are: 'text.md' (484 bytes, documento Markdown, 11 mayo 2021, 18:10), 'solution.sql' (20 bytes, código SQL, 21 mayo 2020, 19:06), 'problem.json' (106 bytes, documento JSON, 11 mayo 2021, 18:40), 'insert.sql' (285 bytes, código SQL, 21 mayo 2020, 19:10), and 'create.sql' (239 bytes, código SQL, 11 mayo 2021, 18:07).

The screenshot shows a file explorer window titled 'dml.zip'. It displays a list of files with columns 'Nombre', 'Tamaño', 'Tipo', and 'Modificado'. The files are: '01_insert_row.zip' (1,5 kB, archivador Zip, 08 junio 2020, 19:34), '02_insert_rows.zip' (1,7 kB, archivador Zip, 08 junio 2020, 19:34), and '03_insert_constraint.zip' (2,0 kB, archivador Zip, 08 junio 2020, 19:34).

Uso durante el curso 2020/21

Uso en «Bases de Datos»

- Una versión preliminar de LearnSQL se utilizó en el grupo de 2ºB de «Bases de Datos» (GII+GIC) en el curso 2020-21
- Principalmente se utilizó para ejercicios de consultas SQL:
 - 39 problemas de consultas SQL (consultas de una tabla, consultas con reuniones, agregaciones y agrupaciones, operaciones de conjuntos y consultas anidadas)
 - 1 problema DML
 - 5 problemas sobre funciones PL/SQL
 - 3 problemas sobre procedimientos PL/SQL
 - 1 problema sobre disparadores
 - 0 problemas discriminantes (*añadidos en abril de 2021*)

Participación de los estudiantes

- En general, creo que se consiguió una **participación bastante aceptable**
- Este grupo consta de 51 estudiantes matriculados:
 - 26 estudiantes realizaron algún envío
 - 26 estudiantes realizaron algún envío correcto
 - El estudiante que menos usó el juez realizó 2 envíos
 - El estudiante que más usó el juez realizó 1126 envíos
 - El número medio de envíos por usuario es de 212 con mediana 144

- Al final del curso se publicó una encuesta anónima que contestaron 11 estudiantes, de los cuales 10 habían intentado más de 20 problemas
- Puntuación obtenida para algunas preguntas (sobre 5)
 - *La organización de las distintas páginas del juez es adecuada:* **4.73**
 - *El juez es fácil de usar:* **4.82**
 - *La retroalimentación que muestra el juez cuando un envío no es correcto es informativa:* **2.73**
 - *El juez automático te ha ayudado a aprender a realizar consultas SQL, en proporción al uso que has hecho de él:* **4.82**
- También se recabaron comentarios libres sobre aspectos negativos y funcionalidades que echaban en falta

Conclusiones

Conclusiones

- Se ha desarrollado un juez automático para diversos ejercicios de «Bases de Datos»
- Se ha probado en un grupo durante el curso 2020-21 con resultados esperanzadores. El sistema ha mejorado mucho desde enero, así que esperamos que el curso que viene sea aún más útil
- El sistema está abierto a cualquier profesor de «Bases de Datos» y similares. **Si estáis interesados, contactad conmigo en emartinm@ucm.es**

- Tenemos bastantes mejoras, extensiones y correcciones apuntadas:
 - <https://github.com/emartinm/lsql/blob/master/TODO.md#tareas-propuestas>
 - <https://github.com/emartinm/lsql/issues>
- ❶ Ampliar la colección de problemas con más problemas de PL/SQL y problemas discriminantes
- ❷ Soportar otros SGBD como PostgreSQL, pero incluso se podrían añadir sistemas NoSQL como MongoDB, Redis, HBase o Neo4J
- ❸ Comprobar la salida estándar de PL/SQL
- ❹ Permitir un uso abierto donde el estudiante escribe consultas y ve el resultado sin tener que resolver ningún problema concreto (similar a *LiveSQL* de Oracle)
- ❺ Personalizar más y mejor la interfaz de administrador para que sea más amigable y usable por parte del profesor

Enrique Martín

Taller LearnSQL

Fac. Informática - UCM

37 / 39

Vuestra opinión es muy importante

- Como docentes, vuestra visión del juez nos resultará muy interesante y seguro que nos ayuda a mejorar el sistema
- Hemos hecho una encuesta online para recoger vuestra opinión. Os agradeceremos mucho si podéis dedicar 3-5 minutos para rellenarla:
 - <https://forms.gle/cCWBQgfnHgMzkDn9>
- **¡Todo comentario o sugerencia será bienvenido!**

Enrique Martín

Taller LearnSQL

Fac. Informática - UCM

38 / 39

¡Muchas gracias por vuestra asistencia!

Enrique Martín

Taller LearnSQL

Fac. Informática - UCM

39 / 39

6.4. Diapositivas utilizadas en la jornada AprendeTIC



Juez automático para el aprendizaje de bases de datos

Jesús Correas, **Enrique Martín**, Manuel Montenegro, Adrián Riesco, Rubén Rubio
Facultad de Informática, UCM

Jornada «Aprendizaje Eficaz con TIC en la UCM», 23 de junio de 2021

Contenidos

- 1 Introducción
- 2 Características de LearnSQL
- 3 Uso durante el curso 2020/21
- 4 Conclusiones

Introducción

- Los **jueces automáticos** son aplicaciones web que albergan una colección de ejercicios para los estudiantes.
- Cada vez que un estudiante envía una solución, el juez la corrige de manera automática y proporciona una **retroalimentación inmediata**, por lo que permite practicar en cualquier momento y desde cualquier lugar.
- Son muy útiles en asignaturas con una importante **carga práctica**, y en la Fac. de Informática los llevamos utilizando desde hace varios años en asignaturas de programación.

Bases de datos

- Las bases de datos son una **materia básica** en cualquier programa de estudios con **perfil tecnológico**.
- Las bases de datos proporcionan un **lenguaje de consulta** para acceder a la información. Además, los sistemas de bases de datos más comunes permiten definir **fragmentos de código** que modifican la base de datos.
- Por lo tanto, el aprendizaje de las bases de datos tiene un componente **fundamentalmente práctico** y se podrían **beneficiar del uso de jueces automáticos**.

Jueces automáticos para bases de datos

Aunque existen algunos jueces automáticos para utilizar en asignaturas de bases de datos, no se han llegado a aplicar ampliamente en nuestra docencia porque:

- Son **sistemas privados**, que no puedes instalar en tu servidor y usar en tu clase.
- Son **sistemas cerrados** en los que no puedes modificar los problemas que contienen.
- Utilizan un **sistema de bases de datos** diferente al que usas en clase (PostgreSQL, MySQL, SQL Server...) lo que introduce ligeras diferencias sintácticas y de funcionamiento que confunden al estudiante.

En el proyecto INNOVA-Docencia nº 18 «*Juez automático para el aprendizaje de bases de datos*» hemos desarrollado **LearnSQL**, un juez automático adaptado a las bases de datos persiguiendo las siguientes metas:

- **Código abierto** en <https://github.com/emartinm/lsql>. Cualquiera puede instalarlo y modificarlo.
- **Aplicable** a la asignatura de «Bases de datos» de la Facultad de Informática sin esfuerzo.
- **Sencillo** de utilizar tanto para profesores como estudiantes.
- Centrado en el **aprendizaje**: dar todas las ayudas para que el estudiante entienda el error en su envío y pueda corregirlo.
- Abierto a la **ludificación**: logros, podios, clasificación...

Equipo de diseño y desarrollo

- 1 Iker Burgoa (estudiante TFG).
- 2 Pablo Cerro (profesor UCM/UAM).
- 3 Jesús Correas (profesor UCM).
- 4 Tamara Huertas (estudiante TFG).
- 5 Daniel Ibáñez (estudiante TFG).
- 6 Enrique Martín (profesor UCM).
- 7 Manuel Montenegro (profesor UCM).
- 8 Rubén Rubio (profesor UCM).
- 9 Iván Ruiz (estudiante TFG).

Características de LearnSQL

Ejemplo de problema (I)

Seleccionar filas

Considera una tabla que almacena algunos datos sobre clubes de fútbol definida de la siguiente manera:

```
CREATE TABLE Club(  
  CIF CHAR(9) PRIMARY KEY,  
  Nombre VARCHAR(40) NOT NULL UNIQUE,  
  Sede VARCHAR(30) NOT NULL,  
  Num_Socios INTEGER NOT NULL,  
);
```

Escribe una consulta SQL que devuelve todos los datos de aquellos clubes que tienen entre **70.000 y 80.000** socios (ambos incluidos). El esquema del resultado debe ser el siguiente:

```
(CIF, Nombre, Sede, Num_Socios)
```

Ejemplo de problema (II)

Base de datos

[Descargar script](#)

CLUB 

CIF	NOMBRE	SEDE	NUM_SOCIOS
11111111X	Real Madrid CF	Concha Espina	70000
11111112X	Futbol Club Barcelona	Aristides Maillol	80000
11111113X	Paris Saint-Germain Football Club	Rue du Commandant Guilhaud	1000

Resultado esperado

CIF	NOMBRE	SEDE	NUM_SOCIOS
11111111X	Real Madrid CF	Concha Espina	70000
11111112X	Futbol Club Barcelona	Aristides Maillol	80000

Retroalimentación detallada: error en ejecución

Cuando se produce un error en ejecución se indica la **razón** y la **línea** y **columna** donde se ha producido ese error:

```
1 SELECT * FROM Club WHERE Num_Socios < 5
```

Enviar solución

Retroalimentación

36

Posición: línea 1, columna 10

ORA-00923: FROM keyword not found where expected

Retroalimentación detallada: resultados incorrectos

Indica exactamente qué filas son incorrectas:

```
1 SELECT *
2 FROM Club
3 WHERE Num_Socios < 75000
```

Enviar solución

Retroalimentación

Existen algunas filas incorrectas. A continuación se muestran todas las filas, remarcando aquellas que contienen valores incorrectos en alguna columna o que no deberían aparecer.

CIF	NOMBRE	SEDE	NUM_SOCIOS
11111111X	Real Madrid CF	Concha Espina	70000
11111113X	Paris Saint-Germain Football Club	Rue du Commandant Guilbaud	1000

Enrique Martín (emartinm@ucm.es)

LearnSQL

23 de junio de 2021

13 / 22

Ludificación: logros

- Los estudiantes consiguen logros al resolver problemas.
- El estudiante puede consultar la página «Mis logros» para ver sus logros pendientes y los que ya ha conseguido:

Página de logros de Eva

Logros conseguidos

Nombre	Descripción	Fecha
Principiante	Resuelve 5 problemas de cualquier tipo	18 de Noviembre de 2020 a las 21:29
Amigo del podio	Entra en el podio en al menos 3 problemas	18 de Noviembre de 2020 a las 21:29

Logros pendientes

Nombre	Descripción
Maestro	Resuelve 25 problemas de cualquier tipo

Enrique Martín (emartinm@ucm.es)

LearnSQL

23 de junio de 2021

14 / 22

Ludificación: clasificación

Para cada grupo de clase y colección de problemas se puede visualizar la clasificación (por número de problemas resueltos y envíos necesarios para resolver dichos problemas):

Pos.	Usuario	EJ1	EJ2	EJ3	EJ4	Puntuación	Resueltos
1	Eva 🏆x4	1/2 (2)	1/5 (5)	1/3 (3)	1/1 (1)	11	4
2	Juan 🏆x4	1/8 (8)	1/1 (1)	1/14 (14)	3/7 (3)	26	4
3	Ana 🏆x4	7/34 (17)	2/32 (15)	3/11 (8)	3/11 (8)	48	4
4	Alberto 🏆x5	1/82 (44)	1/6 (6)	1/91 (91)	1/57 (57)	198	4
5	Jacinto 🏆x1	1/3 (3)	0/0 (0)	1/2 (2)	1/1 (1)	6	3
6	Soledad 🏆x5	1/8 (8)	0/6 (6)	1/2 (2)	1/4 (4)	14	3

La clasificación incluye el número de logros obtenido por cada estudiante al lado de su nombre.

Enrique Martín (emartinm@ucm.es)

LearnSQL

23 de junio de 2021

15 / 22

Uso durante el curso 2020/21

Uso en «Bases de Datos»

- Una versión preliminar de LearnSQL se utilizó en un grupo de «Bases de Datos» en el curso 2020-21.
- De los 51 estudiantes matriculados se consiguió una **participación bastante aceptable**:
 - 26 estudiantes realizaron algún envío.
 - 26 estudiantes realizaron algún envío correcto.
 - El estudiante que menos usó el juez realizó 2 envíos.
 - El estudiante que más usó el juez realizó 1126 envíos.
 - El número medio de envíos por usuario es de 212 con mediana 144.

Valoración de los estudiantes

- Al final del curso se publicó una encuesta anónima que contestaron 11 estudiantes, de los cuales 10 habían intentado más de 20 problemas.
- Puntuación obtenida para algunas preguntas (sobre 5).
 - *La organización de las distintas páginas del juez es adecuada:* **4.73**.
 - *El juez es fácil de usar:* **4.82**.
 - *La retroalimentación que muestra el juez cuando un envío no es correcto es informativa:* **2.73**.
 - *El juez automático te ha ayudado a aprender a realizar consultas SQL, en proporción al uso que has hecho de él:* **4.82**.

Conclusiones

Conclusiones

- Se ha desarrollado un juez automático útil para una asignatura de «Bases de Datos» que soporta diversos tipos de ejercicios. Incorpora elementos de ludificación y proporciona toda la información posible para que el estudiante entienda sus fallos.
- Se ha probado en un grupo durante el curso 2020-21 con resultados esperanzadores. El sistema ha mejorado mucho desde enero, así que esperamos que el curso que viene sea aún más útil.

Trabajo futuro

- 1 Estudiar el impacto real en el aumento del aprendizaje de los estudiantes.
- 2 Ampliar la colección de problemas.
- 3 Soportar otros sistemas de bases de datos como PostgreSQL, e incluso sistemas NoSQL como MongoDB, Redis, HBase o Neo4J.

¡Muchas gracias!

6.5. Encuesta de valoración del juez por parte de los asistentes al taller

Valoración de LearnSQL

Os agradecemos mucho si podéis dedicar unos instantes a rellenar esta encuesta de valoración sobre LearnSQL.
La información es totalmente anónima.

***Obligatorio**

1. ¿Qué dispositivo has utilizado para acceder a LearnSQL?

Marca solo un óvalo.

- ☐ Ordenador de sobremesa o portátil
- ☐ Móvil
- ☐ Tablet
- ☐ No he podido acceder a LearnSQL
- ☐ Otro: _____

2. Valora de 1 a 5 los siguientes aspectos de la funcionalidad LearnSQL

Marca solo un óvalo por fila.

	1 (Muy negativa)	2	3	4	5 (Muy positiva)
Navegación entre las secciones del juez	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Navegación entre los distintos problemas	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Página de clasificación	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Página de 'Mis envíos'	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Página con la descripción del problema e introducción de la solución	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Página de ayuda (disponible únicamente en español)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

3. ¿Cuál es tu valoración global de LearnSQL?

Marca solo un óvalo.

	1	2	3	4	5	
Muy negativa	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muy positiva

4. ¿Qué aspectos positivos de LearnSQL destacarías? ¿Qué funcionalidades consideras útiles?

5. ¿Qué aspectos negativos encuentras en LearnSQL? ¿Qué funcionalidades cambiarías o eliminarías?

Utilidad de LearnSQL

6. ¿Has impartido alguna vez o piensas impartir la asignatura Bases de Datos? *

Marca solo un óvalo.

- ☐ Sí Salta a la pregunta 7
☐ No, pero sí una asignatura relacionada con las bases de datos Salta a la pregunta 8
☐ No Salta a la pregunta 11

LearnSQL y asignatura Bases de Datos

7. ¿Utilizarías LearnSQL en la asignatura Bases de Datos?

Marca solo un óvalo.

- ☐ Sí
☐ No
☐ Tal vez

Salta a la pregunta 11

LearnSQL y otras asignaturas relacionadas con las bases de datos

8. ¿Qué asignatura(s) relacionada(s) con las bases de datos impartes?

9. ¿Utilizarías LearnSQL para esa(s) asignatura(s)?

Marca solo un óvalo.

- ☐ Sí
☐ No
☐ Tal vez

10. Si LearnSQL no es aplicable para tu(s) asignatura(s), ¿qué funcionalidades necesarias para que lo fuese?

Salta a la pregunta 11

Otros comentarios

11. Si lo deseas, indica a continuación cualquier otro comentario o sugerencia que consideres conveniente.

6.6. Resultados de la encuesta de valoración del juez

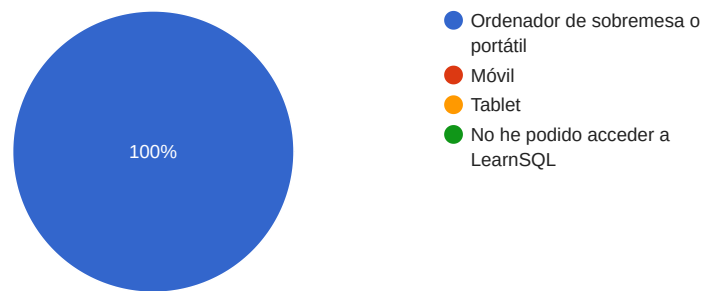
Valoración de LearnSQL

6 respuestas

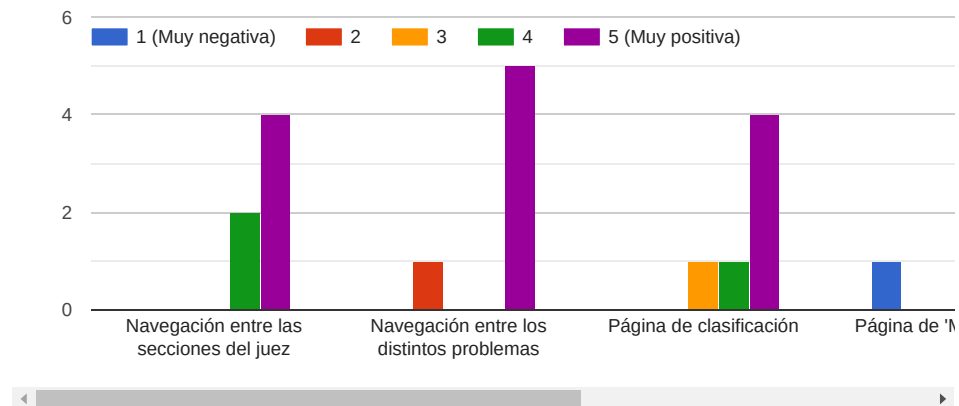
[Publicar datos de análisis](#)

¿Qué dispositivo has utilizado para acceder a LearnSQL?

6 respuestas



Valora de 1 a 5 los siguientes aspectos de la funcionalidad LearnSQL



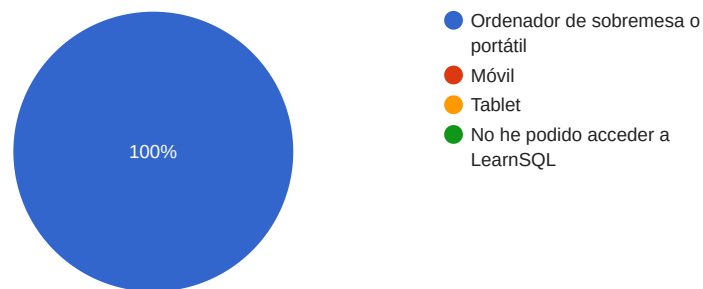
Valoración de LearnSQL

6 respuestas

[Publicar datos de análisis](#)

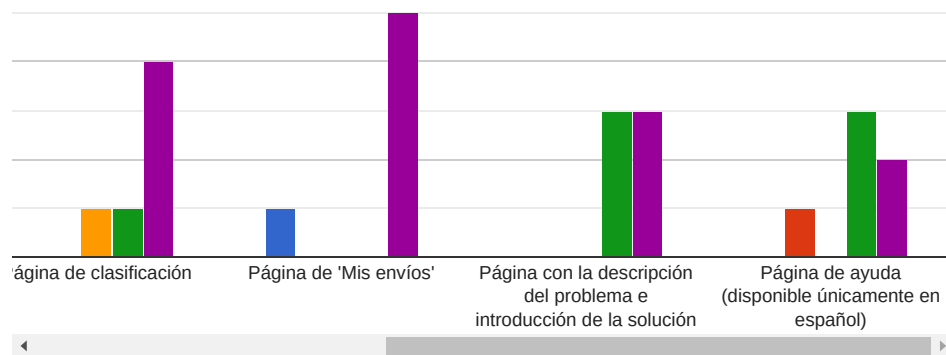
¿Qué dispositivo has utilizado para acceder a LearnSQL?

6 respuestas



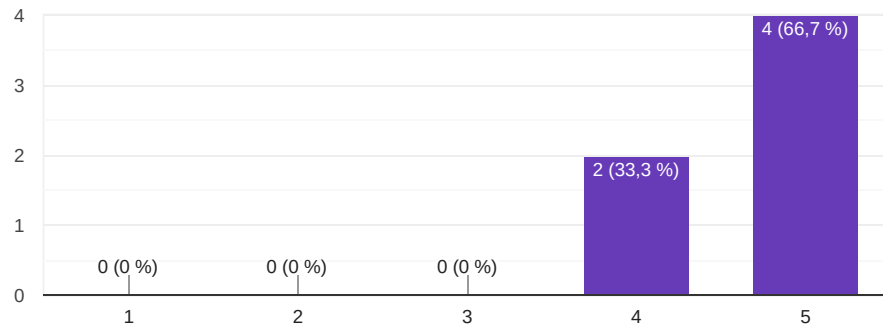
Valora de 1 a 5 los siguientes aspectos de la funcionalidad LearnSQL

(Muy positiva)



¿Cuál es tu valoración global de LearnSQL?

6 respuestas



¿Qué aspectos positivos de LearnSQL destacarías? ¿Qué funcionalidades consideras útiles?

5 respuestas

Atrae al alumno, seguramente mas a los que no tienen un perfil no informático.

Innovación. Evaluación rápida

guía hacia la solución correcta

El sistema funciona, y soporta muchos ejercicios interesantes

La facilidad de uso, la gamificación.



¿Qué aspectos negativos encuentras en LearnSQL? ¿Qué funcionalidades cambiarías o eliminarías?

4 respuestas

Sería interesante que funcionara con Mysql. Además poder ver la solución en varios lenguajes.(por pedir!)

navegación. Se obtiene AC cableando las respuestas. por ejemplo, "arbitros que han arbitrado victoria local" los buscas a mano y te dejas de joins :D comparar contra casos de prueba escondidos evita esto...

que hay que poblarlo de problemas para que cubra todos los aspectos

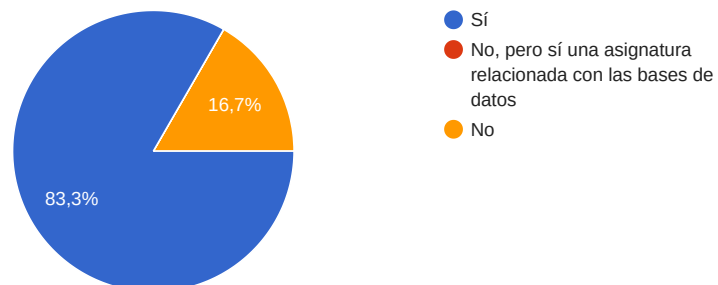
Cosas pequeñas:

- enlaces a una referencia del lenguaje sería útil (estoy *muy* oxidado en SQL, y nunca había usado el de Oracle)
- para ir al siguiente ejercicio de la categoría, una vez resuelto uno, no parece haber un botón de "siguiente".
- en la pantalla de selección de ejercicios, estaría bien ver algo de la forma 10/12 (en sentido resueltos/totales)

Utilidad de LearnSQL

¿Has impartido alguna vez o piensas impartir la asignatura Bases de Datos?

6 respuestas

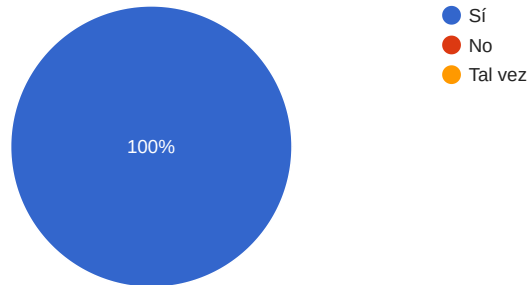


LearnSQL y asignatura Bases de Datos



¿Utilizarías LearnSQL en la asignatura Bases de Datos?

5 respuestas



LearnSQL y otras asignaturas relacionadas con las bases de datos

¿Qué asignatura(s) relacionada(s) con las bases de datos impartes?

0 respuestas

Aún no hay respuestas para esta pregunta.

¿Utilizarías LearnSQL para esa(s) asignatura(s)?

0 respuestas

Aún no hay respuestas para esta pregunta.

Si LearnSQL no es aplicable para tu(s) asignatura(s), ¿qué funcionalidades necesitarías para que lo fuese?

0 respuestas

Aún no hay respuestas para esta pregunta.

Otros comentarios



Si lo deseas, indica a continuación cualquier otro comentario o sugerencia que consideres conveniente.

4 respuestas

Enhorabuena por la herramienta.

Ampliación a PostgreSQL como de complicado sería?
Integración con acepta el reto? te quita el pensar la interfaz

De nuevo, dar la enhorabuena a los que han trabajado en el sistema

Espacios para poder utilizar con distintos grupos de alumnos (aclarar si lo incluye, por favor), de forma que como profesor pueda crear un espacio para mis alumnos. Disponer de un bloque general de problemas que se pueda alimentar de distintas fuentes y utilizar por cada profesor.

Este contenido no ha sido creado ni aprobado por Google. [Notificar uso inadecuado](#) - [Términos del Servicio](#) - [Política de Privacidad](#)

Google Formularios

